FILE NO. 00-077

United States Utility
Patent Application entitled:

DRIVER WAVEFORM MODELING WITH MULTIPLE EFFECTIVE CAPACITANCES

Inventors:

Prasad Subbarao
Sandeep Bhutani
Charutosh Dixit
Prabhakaran Krishnamurthy

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

"Express Mail" Mailing Label Number

EL591659519US

Date of Deposit: March 21, 2001
I hereby certify that this paper is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" Service under 37 CFR §1.10 on the date indicated above and is addressed to the Commissioner for Patents, Washington, D.C. 20231

Pamela M. Stone
(Typed or printed names of person mailing)

Pamela M. Stone
(Signature of person mailing)

DOCKET NO. 00-077
FETF:68513

DRIVER WAVEFORM MODELING WITH MULTIPLE EFFECTIVE CAPACITANCES

BACKGROUND OF THE INVENTION

The present invention relates generally to electronic circuit design using computer simulation techniques. More specifically, but without limitation thereto, the present invention relates to modeling the output waveform of a circuit element driving a distributed resistance-capacitance network.

Methods for modeling the output waveform of a circuit element or cell driving a resistance-capacitance network generally use a Thevenin equivalent model, i.e., a voltage source in series with a resistance, and a load having a single effective capacitance. However, previous methods produced unsatisfactory results using a computer circuit simulation program such as SPICE because the actual waveform delays were found to be more than ten percent larger than those predicted by the simulation program. A method is therefore needed for modeling the output waveform of a cell driving a resistance-capacitance network that compares more closely to the actual output waveform.

SUMMARY OF THE INVENTION

The present invention advantageously addresses the needs above as well as other needs by providing a method for modeling the output waveform of a cell driving

a resistance-capacitance network that includes multiple effective capacitances.

In one embodiment, the invention may be characterized as a method of calculating Thevenin parameters that includes the steps of (a) initializing estimates of effective capacitances *Ceff1* and *Ceff2*, of a switching threshold delay *t0*, and of a slope delay *deltat*; (b) solving ramp response equations for *t0* and *deltat* as a function of *Ceff1* and *Ceff2*; (c) comparing the estimates of *t0* and *deltat* with solutions for *t0* and *deltat* found in step (b); and (d) replacing the estimates of *t0* and *deltat* with the solutions for *t0* and *deltat* if the solutions for *t0* and *deltat* have not converged to the estimates of *t0* and *deltat*.

In another embodiment, the invention may be characterized as a computer program product that includes a medium for embodying a computer program for input to a computer and a computer program embodied in the medium for causing the computer to perform at least one of the following functions:

(a) initializing estimates of effective capacitances *Ceff1* and *Ceff2*, of a switching threshold delay *t0*, and of a slope delay *deltat*;

(b) solving ramp response equations for *t0* and *deltat* as a function of *Ceff1* and *Ceff2*;

(c) comparing the estimates of *t0* and *deltat* with solutions for *t0* and *deltat* found in step (b);

(d) replacing the estimates of *t0* and *deltat* with the solutions for *t0* and *deltat* if the solutions for *t0* and *deltat* have not converged to the estimates of *t0* and *deltat*;

(e) repeating steps (b), (c), and (d) until the solutions for *t0* and *deltat* converge to the estimates of *t0* and *deltat*;

(f) calculating a *delay1* as a function of *t30(Ceff1)* or *t70(Ceff1)* and a *delay2* as a function of *t50(Ceff2)* from a Foster or a pi model;

(g) comparing *delay1* and *delay2* to delays *delay1'* and *delay2'* corresponding to *Ceff1* and *Ceff2* in a delay lookup table;

(h) finding new values for *Ceff1* and *Ceff2* from a reverse lookup of *delay1* and *delay2* in the delay lookup table if the calculated values for *delay1* and *delay2* have not converged to delays *delay1'* and *delay2'*;

(i) replacing the estimates of *Ceff1* and *Ceff2* in step (b) with the new values for *Ceff1* and *Ceff2*; and

(j) repeating steps (b) through (i) until the calculated values for *delay1* and *delay2* converge to delays *delay1'* and *delay2'*.

The features and advantages summarized above in addition to other aspects of the present invention will become more apparent from the description, presented in conjunction with the following drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other aspects, features and advantages of the present invention will be more apparent from the following more specific description thereof, presented in conjunction with the following drawings wherein:

FIG. 1 is a schematic diagram of a Thevenin equivalent model of the prior art;

FIG. 2 is a schematic diagram of a Foster load synthesis model of the prior art for modeling the voltage response of the CMOS gate of FIG. 1;

FIG.3 is a schematic diagram of a pi load synthesis model of the prior art for modeling the voltage response of the CMOS gate of FIG. 1;

FIG. 4 is an ideal voltage plot of the prior art illustrating Thevenin voltage parameter definitions for the CMOS gate of FIG. 1;

FIG. 5 is a voltage plot illustrating typical output voltage response of the CMOS gate of FIG. 1 as a function of time; and

FIG. 6 is a flowchart of an embodiment of the present invention for calculating Thevenin parameters and the effective capacitances *Ceff1* and *Ceff2*.

Corresponding reference characters indicate corresponding elements throughout the several views of the drawings.

DETAILED DESCRIPTION OF THE DRAWINGS

The following description is presented to disclose the currently known best mode for making and using the present invention. The scope of the invention is defined by the claims.

FIG. 1 is a schematic diagram of a Thevenin equivalent model 100 of the prior art. Shown are a Thevenin voltage source 102, a driver resistance 104, a CMOS gate 150 and a load impedance 152. The CMOS gate 150 includes the Thevenin voltage source 102, which is connected in series with the driver resistance 104. The

Thevenin equivalent model 100 is used to model a typical cell in integrated circuit designs.

FIG. 2 is a schematic diagram of a Foster load synthesis model 200 of the prior art for modeling the voltage response of the CMOS gate 150 of FIG. 1. Shown are a driver resistance 104, a first current branch 202, a first resistor 204, a first capacitor 206, a second current branch 208, a second resistor 210, and a second capacitor 212. The Foster load synthesis model 200 is connected in series with the driver resistance 104 of the CMOS gate 150 and includes multiple current branches 202 and 208 connected in parallel. In the first current branch 202, the first resistor 204 is connected in series with the first capacitor 206. Likewise in the second current branch 208, the second resistor 210 is connected in series with the second capacitor 212.

FIG.3 is a schematic diagram of a pi load synthesis model 300 of the prior art for modeling the voltage response of the CMOS gate 150 of FIG. 1. Shown are a driver resistance 104, a first capacitor 302, a first resistor 304, and a second capacitor 306. The pi load synthesis model 300 is connected in series with the driver resistance 104 of the CMOS gate 150. The first capacitor 302 is connected between the driver resistance 104 and ground. The first resistor 304 and the second capacitor 306 are connected in series across the first capacitor 302.

FIG. 4 is an ideal voltage plot vs. time 400 illustrating Thevenin voltage parameter definitions for an inverting CMOS gate. Shown are a gate input signal 402, a Thevenin voltage 404, a switching threshold delay ($t0$)

406, and a slope delay (*deltat*) 408. The delay between the time when the gate input signal 402 reaches the logic switching threshold *Vst* and the time when the Thevenin voltage 404 begins to fall defines the switching threshold delay (*t0*) 406. The delay between the time when the Thevenin voltage 404 begins to fall and the time when the Thevenin voltage 404 reaches zero defines the slope delay (*deltat*) 408. Similar Thevenin voltage parameters may be defined for other CMOS gates and input transitions.

Using the Thevenin model, the output voltage *Vo'* of the CMOS gate 150 may be expressed as a function of time *t* by

$$Vo'(t) \equiv y(t,t0,deltat) \qquad (1)$$

Specifically, for a falling output transition

$$\begin{aligned} y(t,t0,deltat) &= Vdd & t - t0 &< 0 \qquad (2)\\ y(t,t0,deltat) &= Vdd - y0(t,t0,deltat) & 0 \le t - t0 &< deltat\\ y(t,t0,deltat) &= Vdd - y0(t,t0,deltat) + y0(t - deltat,t0,deltat) & t - t0 &\ge deltat \end{aligned}$$

and for a rising output transition

$$\begin{aligned} y(t,t0,deltat) &= 0 & t - t0 &< 0 \qquad (3)\\ y(t,t0,deltat) &= y0(t,t0,deltat) & 0 \le t - t0 &< deltat\\ y(t,t0,deltat) &= y0(t,t0,deltat) - y0(t - deltat,t0,deltat) & t - t0 &\ge deltat \end{aligned}$$

where *y0(u,v,w)* is the single positive ramp response for a capacitive load C and a driver resistance Rd, which may be found from the equation

$$y0(u,v,w) = Vdd \cdot \left[\frac{u - v}{w} - \left(\frac{Rd \cdot C}{w}\left(1 - e^{-\frac{u-v}{Rd \cdot C}}\right)\right)\right] \qquad (4)$$

Equations (1), (2), (3), and (4) are derived and explained in detail in the article "Performance Computation for Precharacterized CMOS Gates with RC Loads", Florentin Dartu, Noel Menezes, and Lawrence T. Pileggi, <u>IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems</u>, Vol. 15, No. 5, May 1996.

FIG. 5 is a voltage plot 500 illustrating typical output voltage response *Vo'(t)* of the CMOS gate 150 of FIG. 1 as a function of time. As the effective capacitance of the load 152 builds up charge, the output voltage *Vo'(t)* increases from zero to 30 percent of the final voltage for a rising transition in a time *t30* from the input switching threshold and continues increasing to 50 percent of the final voltage in a time *t50* measured from the input switching threshold. For a falling transition, the output voltage Vo' drops to 70 percent of the initial voltage in a time *t70* from the input switching threshold.

In contrast to methods that solve for the Thevenin parameters using a single effective capacitance, the following method uses two values of effective capacitance *Ceff1* and *Ceff2*. *Ceff1* is used to find either *t30(Ceff1)* (for a rising transition) or *t70(Ceff1)* (for a falling transition), and *Ceff2* is used to find *t50(Ceff2)* to calculate the Thevenin parameters from the ramp response equations

$$y(t50(Ceff2), t0, deltat) = 0.5 \cdot Vdd \quad (5)$$

$$y(t30(Ceff1), t0, deltat) = 0.3 \cdot Vdd \text{ (or } y(t70(Ceff1), t0, deltat) = 0.7 \cdot Vdd \text{)}$$

Initial estimates are made for *Ceff1*, *Ceff2*, *t0*, and *deltat*, for example:

$$Ceff1 = Ceff2 = .0001\ pf \qquad (6)$$
$$deltat = 5 \cdot t50(Ceff2) - t30(Ceff1)$$
$$t0 = t50(Ceff2) - 0.69 \cdot Rd \cdot Ceff2 - 0.5 \cdot deltat$$

Newton's method may be used to solve the non-linear equations (5) from equations (1) - (4) for *t0* and *deltat* by using the estimates of *Ceff1* and *Ceff2* and a delay lookup table according to well known techniques. The delay lookup table contains a range of delays calculated as a function of input ramptime and input capacitance and is contained in a precharacterized cell library created according to well known techniques. The calculated solutions for *t0* and *deltat* are substituted for the previous estimates of *t0* and *deltat* in the ramp response equations (5) until the solutions for *t0* and *deltat* converge to a desired accuracy.

After computing the Thevenin parameters *t0* and *deltat* from the initial estimates, either the Foster model 200 of FIG. 2 or the pi model 300 of FIG. 3 may be used according to well known techniques such as AWE (asymptotic waveform evaluation) to calculate a first delay, *delay1*, to *t30(Ceff1)* or *t70(Ceff1)* and a second delay, *delay2*, to *t50(Ceff2)*.

The calculated values for *delay1* and *delay2* are compared to corresponding delays *delay1'* and *delay2'* found in the lookup table using the input ramptime and *Ceff1* and *Ceff2*, respectively. If *delay1* and *delay2* have converged to within a desired accuracy of *delay1'* and *delay2'*, then the calculation of the Thevenin parameters *t0* and *deltat*

is complete. If not, then new values for *Ceff1* and *Ceff2* are found by a reverse interpolation of *delay1* and *delay2* from the delay lookup table, i.e.,

$$\begin{aligned} &\textit{delay} = f(\textit{input ramptime, capacitance}) \Rightarrow \\ &\qquad \text{capacitance} = g(\text{input ramptime, delay}) \end{aligned} \tag{7}$$

After obtaining new values for *Ceff1* and *Ceff2*, the Thevenin parameters *t0* and *deltat* are recomputed from the ramp response equations (5), and new values for *delay1* and *delay2* are calculated from the Foster or the pi model until *delay1* and *delay2* converge to *delay1'* and *delay2'* within a desired accuracy, for example, 1 picosecond.

FIG. 6 is a flowchart 600 for calculating the Thevenin parameters of switching threshold delay *t0* and slope delay *deltat* and effective load capacitances *Ceff1* and *Ceff2* according to the method explained above.

Step 602 is the entry point for the flowchart 600.

In step 604, estimates for *Ceff1*, *Ceff2*, *t0*, and *deltat* are initialized.

In step 606, the solutions to the following ramp response equations are calculated:

$y(t50(Ceff2), t0, deltat) = 0.5 \cdot Vdd$ and either

$y(t30(Ceff1), t0, deltat) = 0.3 \cdot Vdd$ or

$y(t70(Ceff1), t0, deltat) = 0.7 \cdot Vdd$ for *t0* and *deltat*.

In step 608, the estimates of *t0* and *deltat* are compared with the calculated solutions. If the solutions for *t0* and *deltat* have not yet converged to the estimates, then the estimates of *t0* and *deltat* are replaced with the calculated solutions in step 610, and step 606 is repeated.

If the solutions for *t0* and *deltat* have converged to the estimates, then the *delay1* is calculated as a function of *t30(Ceff1)* or *t70(Ceff1)* and *delay2* is calculated as a function of *t50(Ceff2)* from a Foster or a pi model in step 612.

In step 614, corresponding delays *delay1'* and *delay2'* for *Ceff1* and *Ceff2* are found in the delay lookup table.

In step 616, the calculated values for *delay1* and *delay2* are compared to the corresponding delays *delay1'* and *delay2'*. If *delay1* and *delay2* have converged to within a desired accuracy of *delay1'* and *delay2'*, then control transfers to step 622.

If *delay1* and *delay2* have not converged to within a desired accuracy of *delay1'* and *delay2'*, then new values for *Ceff1* and *Ceff2* are found from the delay lookup table in step 618.

In step 620, the estimates of *Ceff1* and *Ceff2* are replaced with the new values and step 606 is repeated.

Step 622 is the exit point for the flow chart 600.

The flowchart 600 may also be embodied in a computer program product that includes a medium for embodying a computer program for input to a computer and a computer program embodied in the medium for causing the computer to perform at least one of the following functions:

(a) initializing estimates of effective capacitances *Ceff1* and *Ceff2*, of a switching threshold delay *t0*, and of a slope delay *deltat*;

(b) solving ramp response equations for *t0* and *deltat* as a function of *Ceff1* and *Ceff2*;

(c) comparing the estimates of *t0* and *deltat* with solutions for *t0* and *deltat* found in step (b);

(d) replacing the estimates of *t0* and *deltat* with the solutions for *t0* and *deltat* if the solutions for *t0* and *deltat* have not converged to the estimates of *t0* and *deltat*;

(e) repeating steps (b), (c), and (d) until the solutions for *t0* and *deltat* converge to the estimates of *t0* and *deltat*;

(f) calculating a *delay1* as a function of *t30(Ceff1)* or *t70(Ceff1)* and a *delay2* as a function of *t50(Ceff2)* from a Foster or a pi model;

(g) comparing *delay1* and *delay2* to delays *delay1'* and *delay2'* corresponding to *Ceff1* and *Ceff2* in a delay lookup table;

(h) finding new values for *Ceff1* and *Ceff2* from a reverse lookup of *delay1* and *delay2* in the delay lookup table if the calculated values for *delay1* and *delay2* have not converged to delays *delay1'* and *delay2'*;

(i) replacing the estimates of *Ceff1* and *Ceff2* in step (b) with the new values for *Ceff1* and *Ceff2*; and

(j) repeating steps (b) through (i) until the calculated values for *delay1* and *delay2* converge to delays *delay1'* and *delay2'*.

Other modifications, variations, and arrangements of the present invention may be made in accordance with the above teachings other than as specifically described to practice the invention within the spirit and scope defined by the following claims.